AL/HR-TP-1994-0026

AD-A286 568

ARMSTRONG LABORATORY

TRAINING APPLICATIONS OF NON-DIAGNOSTIC INTELLIGENT TUTORING SYSTEMS

Leo Gugerty

Mei Technology Corporation
8930 Fourwinds Drive
Suite 450
San Antonio, TX 78230

Edward M. Arnold, Jr.

HUMAN RESOURCES DIRECTORATE
TECHNICAL TRAINING RESEARCH DIVISION
7909 Lindbergh Drive
Brooks Air Force Base, TX 78235-5352

September 1994

Interim Technical Paper for Period February 1992 – June 1993

94-35870

Approved for public release; distribution is unlimited.

AIR FORCE MATERIEL COMMAND
BROOKS AIR FORCE BASE, TEXAS

NOTICES

This paper is published as received and has not been edited by the technical editing staff of the Armstrong Laboratory.

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The office of Public Affairs has reviewed this paper, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This paper has been reviewed and is approved for publication.

Edward M. Arnold Jr

EDWARD M. ARNOLD, JR., Captain, USAF
Chief, Research Operations

R Bruce Gould

R. BRUCE GOULD, GS-15
Technical Director
Technical Training Research Division

James B Bushman

JAMES B. BUSHMAN, Lieutenant Colonel, USAF
Chief, Technical Training Research Division

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
September 1994

3. REPORT TYPE AND DATES COVERED
Interim – February 1992 – June 1993

4. TITLE AND SUBTITLE
Training Applications of Non-Diagnostic Intelligent Tutoring Systems

5. FUNDING NUMBERS
C – F33615-D-0651
PE – 62205F
PR – 1121
TA – 11
WU – 04

6. AUTHOR(S)
Leo Gugerty
Edward M. Arnold, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Mei Technology Corporation
8930 Four Winds Drive, Suite 450
San Antonio, TX 78239

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)
Armstrong Laboratory (AFMC)
Human Resources Directorate
Technical Training Research Division
7909 Lindbergh Drive
Brooks Air Force Base, TX 78235-5352

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AL/HR-TP-1994-0026

11. SUPPLEMENTARY NOTES

Armstrong Laboratory Technical Monitor: Edward M. Arnold, Jr., (210) 536-2981.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The keystones of traditional intelligent tutoring systems (ITSs) have been complex procedures for student diagnosis and adaptive instruction based on diagnostic data. While some of these systems have been shown to be effective, they are also very expensive to develop. This paper describes another class of ITSs, non-diagnostic ITSs, which do little or no student diagnosis, and concentrate their intelligence in other areas. Intelligent features of non-diagnostic ITSs include: modeling of expert's reasoning process and cognitive representations (often using graphic displays), comparison of student and expert performance, and replays and summaries of student performance. While traditional, diagnostic ITSs are usually intended to be used in a stand-alone fashion, non-diagnostic tutors are designed to facilitate collaborative learning among students and between teachers and students. The non-diagnostic approach to ITS development offers either a low-cost alternative to traditional ITSs or a way to expand the educational capabilities of traditional systems. This paper presents a framework for comparing the features of non-diagnostic and diagnostic ITSs and data on the costs and educational effectiveness of each type of ITS.

14. SUBJECT TERMS
Computer-based training
Instructional design
Integrated maintenance information system (IMIS)
Non-diagnostic intelligent tutoring system
Training

15. NUMBER OF PAGES
46

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

TABLE OF CONTENTS

PREFACE

Research reported in this paper was conducted in conjunction with a project funded by Armstrong Laboratory, Human Resources Directorate to investigate the training use of a job aid, the Integrated Maintenance Information System (IMIS). This work was accomplished under contract (No. F33615-91-D-0651), with Mei Technology Corporation, San Antonio, TX. Management of this project was provided by the Technical Training Research Division, Instructional Design Branch (AL/HRTC). In addition to the other researchers who worked on this project, we would like to thank Kimberly Hicks, Valerie Shute, Patricia Hsieh, William Walsh, Kevin Singley, and Elizabeth Gibson for comments on this paper and Captain Ed Arnold for the final editing.

SUMMARY

This paper describes alternatives to traditional intelligent tutoring systems (ITSs), called non-diagnostic ITSs. A major advantage of non-diagnostic ITSs is that they may be more useful than traditional ITSs as models of how to add training capabilities to electronic job aids and expert systems.

The keystones of traditional intelligent tutoring systems (ITSs) have been complex procedures for student diagnosis and adaptive instruction based on diagnostic data. While some of these systems have been shown to be effective, they are also very expensive to develop. This paper describes another class of ITSs, non-diagnostic ITSs, which do little or no student diagnosis, and concentrate their intelligence in other areas. Intelligent features of non-diagnostic ITSs include: modeling of experts' reasoning processes and cognitive representations (often using graphic displays), comparison of student and expert performance, and replays and summaries of student performance. While traditional, diagnostic ITSs are usually intended to be used in a stand-alone fashion, non-diagnostic tutors are designed to facilitate collaborative learning among students and between teachers and students.

The non-diagnostic approach to ITS development offers either a low-cost alternative to traditional ITSs or a way to expand the educational capabilities of traditional systems. This paper presents a framework for comparing the features of non-diagnostic and diagnostic tutors. A number of non-diagnostic and diagnostic ITSs are described, and data on the costs and educational effectiveness of each type of ITS is presented. Finally, an example is given of how a maintenance job aid is being converted into a non-diagnostic ITS.

Training Applications of Non-Diagnostic Intelligent Tutoring Systems

I. INTRODUCTION

The use of electronic job aids (or electronic performance support systems) is increasing in the workplace (Gery, 1991). Many of these systems contain job-related knowledge, such as an expert system, that can be used to train workers as well as aid their job performance. If the knowledge, or intelligence, in a job aid is in the proper form, it can be used to develop an intelligent tutoring system (ITS). Traditional ITSs diagnose student strengths and weaknesses and adapt instruction to the knowledge level of individual students. The adaptive instruction provided by these systems has been found to often be quite effective (Merrill, Reiser, Ranney, & Trafton, 1992).

However, using an intelligent job aid as the basis of developing a traditional ITS is usually not possible. This is because of the condition mentioned above, that the knowledge in the job aid must be *in the right form* for it to be useful in a traditional ITS. In order for the knowledge in a job aid to be useful in a traditional ITS, this knowledge must be similar to that used by human experts in the job domain. This type of representation of expert knowledge has been called a *glass-box* system (Anderson, 1988). Many job aids represent job knowledge in a form very different from humans, that is, in a *black-box* system. Black-box experts are very difficult to convert to traditional ITSs because the expert knowledge cannot be used to perform the key task of these systems, student diagnosis.

This paper describes an alternative to traditional ITSs, which I call *non-diagnostic ITSs*. Non-diagnostic ITSs can make use of the job knowledge in black-box job aids and expert systems in ways that traditional ITSs cannot. Thus, non-diagnostic ITSs may provide a good model of how to develop intelligent training systems based on electronic job aids. The main body of the paper describes non-diagnostic ITSs and compares them to traditional ITSs. Following this, an example is given of how a maintenance job aid -- the Integrated Maintenance Information System (Link, Von Holle & Mason, 1987) -- is being developed into a non-diagnostic ITS.

Non-Diagnostic vs. Traditional ITSs

There is a sense in which the goals of traditional ITSs are both too ambitious and too narrow. Most traditional ITSs, such as the List Processor (LISP) Tutor (Anderson & Reiser, 1985), are designed to provide tutoring in a stand-alone setting (i.e., without a human teacher present). This ambitious goal requires that the ITS handle all aspects of the very difficult task of tutoring, including expert problem solving, student diagnosis, tailoring instruction to changing student needs, and providing an instructional environment (e.g., a simulation). On the other hand, the goal of developing very intelligent stand-alone ITSs is narrow in the sense that it limits our conception of how intelligence can be incorporated into computer-based training and education. One key problem focusing on stand-alone ITSs is that we may overlook intelligent computer-based systems that include the teacher as part of the tutorial interaction.

ITSs are currently being developed that break with the pattern of traditional ITSs. An example is the Intelligent Conduct of Fire Trainer (INCOFT), an ITS to train the skill of identifying aircraft from radar displays (Newman, Grignetti, Gross & Massey, 1989). INCOFT does little student modeling and relies on a teacher to provide much of the instruction. Its intelligence lies in its ability to advise students when their performance differs from an expert's, to model experts' reasoning for the student, and to provide useful summaries and replays of student performance that can be discussed by the student and the teacher. Thus, INCOFT acts as an intelligent teacher's aid, and facilitates collaborative learning.

A major goal of this paper is to describe ITSs like INCOFT, and compare these to traditional ITSs like the LISP Tutor. The key features that differentiate INCOFT and the LISP Tutor are student diagnosis and adaptive instruction. The LISP Tutor performs student (or cognitive) diagnosis, that is, it makes inferences about the knowledge and misconceptions underlying student performance. Having a detailed student diagnosis enables the LISP Tutor to adapt its instruction to small changes in student knowledge during a tutoring session. INCOFT, on the other hand, simply records student performance without making inferences about it. Therefore, INCOFT must rely on the teacher to adapt instruction to fine-grained changes in student knowledge. Whether or not an ITS performs student diagnosis has a large effect on how it can be used in instruction. Therefore, I will refer to systems like the LISP Tutor as *diagnostic* ITSs. This term is intended as shorthand for a system that performs both detailed diagnosis and adaptive instruction based on the diagnosis. Systems like INCOFT will be referred to as *non-diagnostic* ITSs (and sometimes as intelligent teacher's aids).

This paper will describe specific features, advantages, and disadvantages of both non-diagnostic and diagnostic ITSs, and estimate the cost of each approach in terms of type and level of development work. This overview should allow someone considering developing or using an ITS to understand the costs and benefits of each approach. Following this comparison, I will show how some computer-based training systems not commonly thought of as ITSs fit into the category of non-diagnostic ITSs. This analysis may help widen our conception of how intelligence can be used in computer-based training and education. Finally, the process of converting the IMIS maintenance job aid into a non-diagnostic ITS will be described.

Rationale for Non-Diagnostic Tutors

Before comparing non-diagnostic and diagnostic ITSs, I will discuss some of the reasons for exploring non-diagnostic tutors. The first reason concerns the difficulty of the tutoring task. Consider the variety of knowledge and skills necessary for tutoring. First, in terms of content knowledge, a tutor must be able to solve problems in the task domain at an expert level, explain the reasoning processes used to obtain these solutions, understand common student misconceptions, and sequence topics so that they build on one another. Second, in terms of tutoring methods, a tutor needs knowledge of high-level teaching strategies such as direct instruction or inquiry learning; lower-level teaching transactions (or tactics) such as questions, hints, explanations, and demonstrations; diagnosis and assessment techniques; and instructional management skills. This last skill is especially difficult to master because it involves using student diagnoses to adjust teaching strategies and transactions in the midst of a highly interactive tutoring session.

In addition, computer-based tutoring systems are at a disadvantage when compared to human tutors because computers have access to a lower bandwidth of information about the student (VanLehn, 1988). For example, computers cannot interact with students in natural language, or perceive emotions or gestures.

Given the impressive array of knowledge and skills required for tutoring and the low bandwidth of information available to computers, it is little wonder that no current ITS can perform all tutoring tasks well. Recently, some ITS researchers have suggested that some tutoring tasks, such as student diagnosis, will require long-term basic research before solutions are found (Burger & DeSoi, 1992).

A second reason for developing non-diagnostic ITSs is that augmenting a teacher's knowledge with a non-diagnostic intelligent teacher's aid may provide just as much and perhaps more educational benefit as replacing a teacher (or at least part of a teacher's task) with a stand-alone, diagnostic ITS. Two trends in the way computers are used in industry and education support this conclusion. First, the distinction between stand-alone ITSs and intelligent teaching aids mirrors one in the field of artificial intelligence, where early expert systems were built with the goal of being stand-alone decision makers, replacing a human. Later expert systems were seen more as decision aids, providing intelligent information to a person who makes the final decision (Woods, Roth & Bennett, 1991). Second, it is arguable that the most effective uses of computers in elementary and secondary education involve software tools -- such as word processors, programming languages, databases, and spreadsheets -- that require extensive teacher guidance and *student collaboration*, and *not stand-alone computer-assisted* instruction (Riel, Levin & Miller-Souviney, 1987). The rest of this article points out some of the educational advantages of non-diagnostic tutors.

A third reason for considering non-diagnostic ITSs is that traditional diagnostic ITSs are very expensive to develop and are applicable only in narrow domains. The LISP Tutor and Sherlock I (Lajoie & Lesgold, 1989), which trains aircraft maintenance skills, are examples of successful, traditional ITSs with these two drawbacks. Non-diagnostic tutors can cut the cost of tutor development by eliminating the need for some of the complex components of traditional ITSs.

Another potential solution to the problem of expensive domain-specific tutors is to develop generic authoring shells to lower the cost of the ITS development process. Examples of shells include the Microcomputer Intelligence for Technical Training Writer (MITTWriter) (Wiederholt, Browning, Norton & Johnson, 1991), the Rapid Prototyping ITS Development System (RAPIDS) (Towne & Munro, 1992), and TERTL (Anderson, Corbett, Fincham, Hoffman & Pelletier, 1992). However, given the educational potential of non-diagnostic ITSs, authoring shells will offer only a limited solution if they are used only to develop additional stand-alone diagnostic tutors.

Table 1. Capabilities of Diagnostic and Non-Diagnostic Intelligent Tutoring Systems

Key: Diagnostic Systems: LISP Tutor (L), Sherlock I (S); Non-Diagnostic Systems: INCOFT (I), MACH III (M)

ITS Module & Capabilities	Low-Tech		⟵⟶		High-Tech	
Expert Module						
Simulates human thought processes?	no expert module		black-box expert	I,M,S	glass-box expert	L
Generates problem solutions on-line?	no expert module		canned solutions for pre-selected problems, or generated solutions based on a simple algorithm	I,M, S	solutions generated for a wide variety of domain problems	L
Diagnostic Module						
Info in student model	no student model / performance history	I,M	specific knowledge & skills based on a black-box expert (e.g., issue-based tutoring)	S	specific knowledge, skills, and mis-conceptions based on a glass-box expert (e.g., model tracing)	L
Use of student model	no student model / performance employs and summaries	I,M			to choose problem/topic sequence, knowledge/skills to coach, and level of coaching	L,S
Instructional Module						
How is the content and sequence of topics or problems determined?	by teacher	I,M	preset by tutor developers		generated on-line	L,S
How is the method of instruction determined?	by teacher	I,M	preset by tutor developers	L,S	generated on-line	
How is the content and timing of instructional interventions determined?	by teacher	I,M	preset by tutor developers	L,S	generated on-line	L,S
Group or individual use?	collaborative	I,M	stand alone	L,S	both	
Interface						
Simulates real-world task context?	not at all		to some extent	M,S	closely	I,L
Models expert reasoning and representations?	very little		somewhat	L,S	extensively	I,M

II. COMPARISON OF DIAGNOSTIC AND NON-DIAGNOSTIC ITSs

Table 1 contains a list of some of the key capabilities or features of ITSs. The table is organized in terms of the four components of traditional ITSs, the expert, diagnosis, instructional, and interface modules (Burns & Capps, 1988). For each capability in the table, a range of options is presented, from "high-tech" options that rely on the computer to perform the pedagogical function (e.g., diagnosis), to "low-tech" options that rely on the teacher (or other students) to perform the function. The table shows the capabilities of two diagnostic ITSs, the LISP Tutor and Sherlock I, and two non-diagnostic ITSs, INCOFT and the Maintenance Aid Computer Hawk Intelligent Institutional Instructor (MACH III) (Kurland, Granville & MacLaughlin, 1992). The LISP Tutor has primarily high-tech features. Sherlock I is less sophisticated than the LISP Tutor, but still possesses the essential capabilities of a diagnostic ITS. The two non-diagnostic ITSs have primarily low-tech features, except in the case of their interfaces, which use high-tech features such as realistic simulation and modeling expert reasoning and representations.

A few points should be made about Table 1. First, the terms high-tech and low-tech are not meant to connote a value judgment. Complex technology is not always the best technology. In fact, solving problems via complex technology is sometimes less effective than simpler technologies, especially when the complex technology excludes people from the decision process and the simpler technology does not (Schumacher, 1973). Second, there is not a strict correspondence between diagnostic ITSs and high-tech features, on the one hand, and non-diagnostic ITSs and low-tech features, on the other. The two non-diagnostic ITSs use some high-tech features, as the table shows. Also, diagnostic ITSs can incorporate low-tech features that facilitate teacher involvement, as do non-diagnostic systems. An example of this is the Sherlock II maintenance skills tutor, which provides precise student diagnosis and adaptive instruction as well as feedback (such as replays and summaries of student performance) intended to foster collaborative learning (Katz & Lesgold, in press). Sherlock II can be considered a hybrid of a diagnostic and a non-diagnostic ITS. (See Appendix A).

The following, first describes each of the ITS capabilities in the table, using examples from various ITSs to explain the differences between low-tech and high-tech options. Then, the diagnostic and non-diagnostic ITSs in Table 1 are compared in terms of the table features.

Expert Module

As Table 1 notes, an important question concerning the expert module is whether it simulates human thought processes. *Black-box* expert modules solve problems using methods completely unlike humans, while *glass-box* experts attempt to simulate the important human thought processes used in the task being instructed (Burton & Brown, 1982). An example of a black-box expert is the early version of the SOPHIE electronics tutor, which used mathematical equations to solve electronics problems. The LISP tutor is an example of a glass-box expert module (Anderson, 1988). This tutor uses hundreds of production (if-then) rules to represent the knowledge and strategies used in LISP programming in a detailed manner. Research has shown

this particular production-rule representation to model human knowledge of LISP fairly well (Anderson, Conrad & Corbett, 1989).

The main advantage of a glass-box model is that its detailed model of human thought processes allows it to more specifically and accurately diagnose student knowledge and misconceptions, and then base instruction (e.g., explanations) on specific student weaknesses. The main disadvantage of glass-box models is their cost. The expert module for the LISP tutor is based on Anderson's ACT* theory of human learning and problem-solving, which is based on years of research and theoretical work (Anderson, 1983).

A second important question characterizing expert modules is whether they generate the steps to solving problems on-line, when presented with a brief problem description, or have the specific solution steps pre-stored in their memory. A system that generates problem solutions on-line usually can solve a wider variety of problems than a system that relies on "canned" (pre-stored) problem solutions. The Integrated Maintenance Training Simulator (IMTS) and its successor, RAPIDS, are examples of ITSs that can generate on-line solutions in their domain (troubleshooting equipment) based on a "deep" model of the structure and behavior of a piece of equipment and a model of general troubleshooting behavior (Towne & Munro, 1988, 1991). Once a model of a piece of equipment is authored into RAPIDS, it can predict the effects of any fault in the system and coach troubleshooting procedures for that fault, even though the effects and troubleshooting procedures for specific faults are not stored in the system.

Sherlock I, another system for training maintenance skills, uses a more low-tech approach to solving problems than IMTS/RAPIDS (Lajoie & Lesgold, 1989). For each problem that it uses, Sherlock pre-stores the outcomes of most of the likely problem-solving actions for that problem. The pre-stored problem-solving outcomes are based on a careful task analysis of the problem solving behaviors and thought processes of experienced and inexperienced problem solvers in the domain of interest (Means & Gott, 1988).

Although conducting a task analysis is time consuming and requires some specialized knowledge, this is a less difficult task than developing a model of problem solving that can generate solutions to arbitrary problems in a domain. Thus, the use of task analysis and pre-stored problem solutions is less expensive and more widely applicable than developing a system that can generate solutions on-line.

Diagnostic Module

The second major component of an ITS, the diagnostic module, allows the system to create student models that record aspects of individual students' performance and knowledge. The ITS then uses information in the student model to tailor its instruction to the needs of individual students.

The most advanced diagnostic modules use performance data concerning the actions students take during problem solving and/or the final results of their problem solving to make

inferences about the knowledge and skills behind individual students' performance. A powerful method for making these inferences, called *model tracing*, can be used if an ITS has a glass-box expert module that models human thinking. Model tracing is used in the LISP tutor. As the student uses the computer to plan and write computer programs, the diagnostic module matches each problem solving action taken by the student with the specific knowledge (i.e., production rules) that the expert module would use to produce that action. The diagnostic module also contains production rules to represent specific student misconceptions, so that when students make errors, it can match them with the underlying misconception. The diagnostic module can then record in the student model production rules that a student knows well, rules the student knows less well, and misconceptions.

The detailed information in a student model created by model tracing can be used by the instructional module in a number of ways, such as in determining the contents of hints and explanations and in selecting problems for students. For example, if the diagnostic module infers that a student mistake is based on a production rule the student knows fairly well the instruction module can give only a general hint to the student. On the other hand, if the student's mistake is based on a production rule the student knows poorly, the instruction module can give a detailed explanation of the mistake and the correct move.

A slightly more low-tech method of student diagnosis is *issue-based tutoring* (Burton & Brown, 1982). An issue-based tutor makes inferences about the knowledge underlying student performance, like a model-tracing tutor. However, issue-based diagnosis can be accomplished with a black-box expert module, whereas model-tracing requires a glass-box expert.

In issue-based diagnosis, each problem solving action that a student or expert could take has associated with it a list of issues (pieces of knowledge) that are required for that action. For example, the WEST system tutors students in how to play an arithmetic board game. Students are repeatedly given three randomly chosen numbers and must use arithmetic operators and other game strategies to combine these numbers so as to move a game piece along a path towards a goal. So, a student move of "(3x2)+1" would be analyzed as using the issues *times, plus, and parentheses.* The diagnostic module would compare this move to the expert's move and increment all those issues the student used, while decrementing those issues used on the expert's move but not the student's. The information in an issue-based student model can be used to tailor the ITS's instruction to specific student knowledge deficits (e.g., parentheses) as in a model-tracing tutor.

The least sophisticated diagnostic modules record only data about student performance in a student model, making no inferences about the knowledge underlying this performance. An example is INCOFT, which monitors and records the aircraft-identification actions students take while they observe radar displays. It also records the timing of students' actions. These data are used by the instructional module in two ways: to provide replays of a problem in which differences between the student's and an expert's performance are pointed out; and to create summaries of student and expert performance on a problem. INCOFT does not use the student

model[1] to adjust the instruction based on a student's performance or to select problems. These tasks are left up to the teacher.

In the extreme case, some of the computer-based training systems discussed in this article record no student performance data, and do no student diagnosis.

Since the student model created by model tracing relies on a glass-box expert that closely simulates human thought procedures, this kind of diagnostic capability is expensive and time consuming to develop. The minimal student model used by INCOFT is obviously much easier to develop. The effort required to develop issue-based student models varies widely depending on the complexity of the issues and the complexity of the schemes by which issues are updated. The student modeling for WEST is relatively simple, compared to a model-tracing tutor. However, developing more complex student modeling schemes using an issue-based approach, such as in the Sherlock II maintenance skill tutor (Katz & Lesgold, 1992), can require extensive cognitive task analysis using domain experts.

Instructional Module

The third major component of an ITS is the instructional module. The problem of delivering and managing instruction via an ITS has received less systematic analysis and development than the problems of expert knowledge and student diagnosis (Pirolli, 1991). Perhaps this is because planning and delivering tutorial instruction is such a complex, interactive task. The decisions a tutor must make include: 1) curricular decisions regarding the content and sequencing of topics or problems, and 2) instructional decisions regarding the type of instructional intervention, the content and timing of instructional interventions, and the overall method of instruction (Halff, 1988; Woolf, 1991). The tutor must choose from a variety of instructional interventions, such as exposition (e.g., explanations, examples of concepts, modeling of procedures), coaching (e.g., hints and explanations during problem solving), and asking and answering questions. Methods of instruction also vary widely, including direct instruction, guided discovery learning, and Socratic dialog. In addition, the advantage, and challenge, of tutorial interaction is that all of these decisions can be changed frequently based on the tutor's assessment of the student's progress, motivation, and learning style.

As Table 1 shows, an ITS can make these curricular and instructional decisions on-line (during a tutorial interaction) using a comparison of the student's and the expert's knowledge states. Alternatively, an ITS's developers could make some or all of these decisions on a one-time basis and hardwire these decisions into the ITS's algorithm. Finally the ITS could leave curricular/instructional decisions up to the teacher.

The first curricular/instructional decision shown in Table 1 focuses on curricular decisions, such as problem sequencing. A number of tutors, including the LISP Tutor, Sherlock,

---

[1] Calling the performance data recorded by INCOFT a "student model" is stretching the definition of this term. Some would restrict the use of the term "student model" to cases where inferences about the knowledge and skills underlying student performance are recorded.

and the BIP-II programming tutor (Halff, 1988), choose problems on-line based on diagnosis of individual students' knowledge. At the other extreme, ITSs like INCOFT require the teacher to make these decisions.

The next instructional decision shown in the table concerns the overall methods of instruction. Woolf (1991) and Shute (1992) have been investigating ITSs that can switch instructional methods based on input from the student. However, almost all existing diagnostic ITSs have a single method of instruction that is used consistently. For example, the LISP tutor uses a directive, problem-based method of instruction. Most instruction is given in the context of problem solving and takes the form of modeling, hints, and explanations. Students are given immediate feedback after errors. Non-diagnostic ITSs like INCOFT and MACH III rely on the teacher to determine the method of instruction.

In terms of more specific decisions about the content and timing of instructional interventions, most diagnostic ITSs make some decisions on-line, while other decisions are preset by the developers, as is shown in the table. For example, in the LISP Tutor, the content of specific hints and explanations was preset by the developers. However, the tutor makes a number of instructional decisions on-line, such as when to intervene (based on student errors), whether to provide a general hint or a detailed explanation (based on the number of student errors or the student's request), and the topic of the hint or explanation (based on the diagnostic module's assessment of the missing knowledge or misconception underlying the student's error). Again, with non-diagnostic ITSs, the teacher must decide what instructional interventions to use, for example, how to use the replays and summaries of student performance.

Finally, Table 1 also characterizes the instructional modules of ITSs according to whether they focus on collaborative or stand-alone use.

To summarize this subsection, even though the LISP tutor is one of the most intelligent of ITSs, the instructional decisions that it generates on-line are based on fairly simple algorithms. This is typical of other diagnostic ITSs. Much of the intelligence of the LISP tutor, and most diagnostic ITSs, lies in the diagnostic and expert modules. For most ITSs, many important curricular/instructional decisions, such as the overall instructional method and the content of explanations, are made on a one time basis by the system developer and cannot be changed by the tutor itself during operation or by the teacher.

Developing ITSs that can flexibly make difficult curricular/instructional decisions is a long term research goal. Progress is being made in this area. Examples include Woolf's (1991) and Shute's (1992) work on ITSs that can switch methods of instruction on-line. Acker, Lester, Souther, and Porter (1991) are investigating how to generate explanations interactively in response to students' questions. The approach taken by non-diagnostic ITSs is to leave difficult curricular/instructional decisions up to the teacher.

Human-Computer Interface

The final ITS component in Table 1 is the interface. Two factors that distinguish low-tech and high-tech interfaces are whether the tutor simulates the real-world task context, and whether the interface allows students to use experts' reasoning and knowledge representations while using the tutor. A realistic simulation environment can help students transfer knowledge from the tutorial to a job situation. Non-diagnostic tutors concentrate their intelligence in the interface. For example, INCOFT uses a realistic, simulated radar display that allows students to solve aircraft identification problems in real time. The artificial intelligence (AI) and psychological expertise required to build a realistic simulation often is less extensive than that needed to create glass-box expert and diagnostic modules. On the other hand, expertise in computer graphics and video is needed, and a thorough task analysis must be done.

Burton (1988) and Bonar (1991) have suggested that the effectiveness of an ITS will be greatly improved if the interface allows students to see and work with experts' reasoning and representations while solving problems. A good example of this capability can be seen in Animate, a tutoring system for algebra word problems (Nathan, Johl, Kintsch & Lewis; 1989; Nathan, Kintsch & Young, 1992). Animate is based on a theory of solving algebra word problems, which holds that two key aspects of experts' representations of word problems are a situation model of the semantics of a problem, and a problem model describing the formal, mathematical relations in the problem. The situation model is represented in Animate by an animated diagram of the problem. The problem model is represented by a graphical network of equations. Students build the animation diagram and the equation network, run the animation, and then make changes in the network or animation until the problem is solved. Research has shown that Animate does improve students' performance on word problems, and that the graphical animation is critical to this improvement (Nathan et al., 1992).

As in constructing a simulation environment, extensive AI knowledge is not required to build an interface like this. Rather, one needs a careful analysis of experts' problem solving processes for the task to be tutored.

Comparing the Capabilities of Diagnostic and Non-Diagnostic ITSs

To summarize the discussion of the capabilities of diagnostic and non-diagnostic ITSs, I will compare four ITSs on each of the capabilities in Table 1. These ITSs are the LISP Tutor, Sherlock I, INCOFT, and MACH II. As the table shows, the LISP Tutor is a diagnostic ITS. It uses high-tech approaches in its expert and diagnostic module, that is, a glass-box expert and model tracing. Its instructional module uses a mixture of high-tech techniques (e.g., choosing the topics and level of detail of hints and explanations on-line) and some less sophisticated ones (using only a single, preset method of instruction). The LISP Tutor's interface simulates real-world programming interfaces closely, and sometimes allows students to use expert task representations (e.g., by showing students templates of LISP functions to fill in).

Sherlock is classified as a diagnostic ITS because its diagnostic module makes inferences about the knowledge and skills underlying student performance. These inferences are based on a fairly detailed and accurate representation of expert troubleshooting knowledge and skill. Also, Sherlock's instructional module is able to generate adaptive responses to student actions on-line. However, in some respects, Sherlock I's capabilities are less sophisticated than the LISP Tutor. For example, Sherlock's expert module uses canned solutions to pre-selected problems, and its diagnostic module uses a complex version of issue-based tutoring.

INCOFT takes a non-diagnostic approach. Although its expert module generates problem solutions on-line, it does this using a simple algorithm. INCOFT's diagnostic module records only performance data about the nature and timing of student responses. The tutor's instructional output consists of replays and summaries of students' performance, and demonstrations of expert performance. These were designed to be used more as informational aids for teachers and students than as stand-alone instructional interventions. INCOFT leaves the decision about how to use these aids, and most other curricular/instructional decisions, up to the teacher. The strength of INCOFT lies in allowing students to practice a real-time task on a realistic interface, and then, via replays and summaries, providing students with comparisons of their performance and that of experts. The important instruction with INCOFT occurs when the teacher and student (or groups of students) discuss the student's replayed problems. While using the replays and summaries, students' do not have the pressure of real-time performance, and can evaluate and discuss their performance.

MACH III is another non-diagnostic ITS, which trains technicians to troubleshoot an Army radar system. Like INCOFT, MACH III does little student modeling, keeping only a record of actions students take during troubleshooting. This record is not used to determine instructional interventions during problem solving. Instead it is used to replay students' performance for a problem. Also like INCOFT, MACH III is designed to be used by students in a classroom under close supervision by a teacher.

MACH III represents expert troubleshooting knowledge in terms of "troubleshooting trees", which show all the general and specific faults that could cause a particular symptom in the radar, as well as the troubleshooting tests to conduct for each specific fault. Students can use MACH III in a straight simulation mode, in which they troubleshoot simulated radar problems without instructional feedback. Alternatively, they can receive feedback (such as why a certain test is recommended) based on the troubleshooting tree, or view the appropriate tree directly (on the computer or a classroom poster). In deciding what feedback to give, MACH III makes no inferences about students' troubleshooting knowledge. Instead, it compares a student's action with the action an expert would make, as shown by the troubleshooting tree.

Most aspects of instruction with MACH III are determined by the teacher, including choosing the overall method of instruction (e.g., simulation with or without feedback), choosing problems, and giving detailed explanations. One advantage of MACH III 's interface is that it shows students expert representations of troubleshooting problems in the form of the troubleshooting trees.

So far this paper examined the differing capabilities of diagnostic and non-diagnostic ITSs, and given a rough indication of the level of effort these capabilities require to develop. Diagnostic tutors are more sophisticated in how they model students' knowledge states and adapt instruction to students' needs. Non-diagnostic tutors focus their intelligence on modeling experts' task knowledge and providing replays and summaries that can facilitate collaborative instruction and learning. Because of the difficulty of developing student diagnosis schemes, diagnostic tutors are usually more costly to develop. A key question for someone who is contemplating developing an ITS is whether the added sophistication of diagnostic ITSs is worth the cost. To begin to answer this, we need to examine some data on the effectiveness of diagnostic and non-diagnostic tutors.

Comparing the Effectiveness of Diagnostic and Non-Diagnostic ITSs

One of the examples typically cited as a diagnostic tutor has been the LISP tutor. The model-tracing approach used in this tutor has also been used in tutors for geometry, algebra, and calculus (Merrill et al., 1992). Anderson et al. (1985) found that students using the LISP tutor took 15.0 hours to complete a set of programming exercises, much faster than students who completed the exercises on their own (26.5 hours), and almost as fast as human-tutored students (11.4 hours). Each group performed equally well on posttests of their programming knowledge. Other model-tracing ITSs, such as Anderson's Geometry Tutor and the Graphical-Instruction-In-LISP (GIL) Tutor, have also been found to be more effective than traditional instruction (Merrill et al., 1992).

Two points should be made about these findings. First, in all of these evaluation studies, students using ITSs also received classroom instruction from a teacher. Thus these studies suggest that model-tracing tutors are effective in outside-the-classroom situations. The studies do not suggest that these tutors can replace human teachers altogether.

The second point concerning the effectiveness of model-tracing tutors is that some of this effectiveness may be due to other aspects of the tutors besides model-tracing, much as the structured editor in the LISP tutor and the graphic interfaces in the Geometry Tutor and GIL. However, studies have shown that when model-tracing diagnosis and its associated instructional guidance are removed from the LISP tutor and GIL, students learn slower and sometimes perform worse than with the full versions of these tutors (Corbett & Anderson, 1991; Merrill et al., 1992). Another study compared a version of GIL that provided very little instructional feedback (that is, where model tracing diagnoses were used only to point out when students made errors) to versions that gave more detailed explanations of the locations of and reasons for errors (Merrill et al., 1992). The versions with more detailed instructional feedback resulted in faster and better student learning. These studies suggest that each of the key aspects of intelligence in a model-tracing ITS--model tracing diagnoses and adaptive instructional feedback based on these diagnoses--can lead to increments in students' learning.

On the other hand, the intelligent capabilities of model-tracing tutors do not always lead to better learning. For example, students using the version of GIL without model-tracing

diagnoses or instructional feedback performed better on a debugging posttest than students using the full-fledged GIL (Merrill et al., 1992). Thus, the detailed and immediate feedback characteristic of model-tracing tutors may deprive students of the opportunity to make, and learn to correct, errors. This issue deserves further investigation, since debugging is an important aspect of programming skill.[2]

Turning now to the second example of a diagnostic ITS, two studies have shown Sherlock I to be very effective. Both studies used pre-test/post-test designs with control groups. Lajoie and Lesgold (1989) found that on a post-test of realistic troubleshooting problems, maintenance trainees with 20 hours of practice using Sherlock made twice as many expert-like moves and half as many inappropriate moves as a control group that received no extra instruction. Another study, reported in Lesgold, Eggan, Katz, and Rao (1992), found that 25 hours of instruction with Sherlock enabled trainees to solve troubleshooting problems at the level of technicians with four additional years of experience. In addition, 90% of these gains were retained after a 6-month delay.

Few studies have been conducted on the effectiveness of non-diagnostic tutors, as these tutors have been developed more recently than diagnostic tutors. INCOFT was not evaluated formally. However, a controlled study of the effectiveness of MACH III has been completed (Acchione-Noel, Saia, Williams & Sarli, 1990). In keeping with MACH III's intended use as a classroom teaching aid, the study compared the use of MACH III with the traditional classroom methods of practicing troubleshooting in a radar maintenance class. The traditional methods involved using procedure manuals and schematics (paper-based practice). Both the MACH III and the "paper-based" group also received lectures and practice on the actual radar equipment.

Although the MACH III students did not perform any better than the paper-based group on practical and written troubleshooting posttests, the tutor students did perform more consistently (i.e., with lower variability). Also, the MACH III group solved significantly more, and more difficult, troubleshooting problems during the class than the paper-based group.

The lack of significant differences in student posttest performance in this initial study should not be taken as a general criticism of non-diagnostic tutors, for a number of reasons. First, the instructors felt they needed more training on how to use MACH III in the classroom. Second, the instructors tended not to use some of the more advanced features of MACH III, such as the troubleshooting trees, because they thought these gave students too much help. The Army school where MACH III was tested (Ft. Bliss) has continued to use the tutor in classes following the tests (Kurland et al., 1992).

---

2 It should be noted that detailed, immediate feedback is not an inherent part of model-tracing tutors. Version of these tutors could be constructed that use detailed, immediate feedback only part of the time. For example, immediate feedback could be discontinued as students became more experienced.

Because MACH III was used in a different instructional context, the evaluation of this ITS cannot be compared easily to the evaluations of diagnostic tutors. The MACH III evaluation studied tutor use in the classroom and used a control group that received extensive classroom instruction. The latter studies (Anderson & Reiser, 1985; Lajoie & Lesgold, 1989) looked at stand-alone ITS use, and found that students using these ITSs performed better than students who received no additional instruction.

The different instructional context used with non-diagnostic ITSs raises questions beyond how to evaluate these tutors. One must consider implementation questions, such as how to integrate these ITSs into the classroom and train instructors to use them. The importance of these questions was highlighted by the instructors in the MACH III study, who asked for more ITS training and tailored the ITS use to their instructional goals in ways not intended by the developers. Questions of teacher training and classroom integration will be considered in the conclusion.

III. CHARACTERISTICS OF NON-DIAGNOSTIC ITSs

The remainder of this paper discusses the capabilities of non-diagnostic ITSs in more detail. Examples of other non-diagnostic tutors will be provided in addition to INCOFT and MACH III, as well as examples of other computer-based training systems that incorporate elements of non-diagnostic tutors. Finally, as an example of how the non-diagnostic approach can be applied, this paper will outline how this approach is being used to convert an expert system for maintenance aiding to a training system.

The characteristics of cognitive apprenticeship training will be used to organize the discussion of the capabilities of non-diagnostic tutors (Collins, Brown & Newman, 1989). Cognitive apprenticeship training pulls together a number of instructional methods that are important for teaching complex skills. These methods include: explicitly *modeling* experts' reasoning processes and representations, *coaching*, *fading*, *articulation/reflection*, and *sequencing* instruction.

Modeling

A key method by which non-diagnostic ITSs provide instruction is to explicitly model experts' reasoning processes and representations. Often, the expert reasoning or representation is displayed graphically in the interface to the ITS, as was done in the Animate word-problem tutor described above. Another tutor that uses graphical displays of experts' representations is MACH III. For example, when making a troubleshooting test, MACH III users can display a troubleshooting tree that explains the reasons for making the test in terms of a hierarchy of faults and symptoms related to the test.

The Explicit Planning and Instantiation by Computer (EPIC) ITS provides textual displays of experts' representations (Twidale, 1989). This ITS tutors students in how to create

proofs in propositional calculus. It provides templates of expert plans for proof construction, and heuristics for selecting plans. Also, its interface helps students keep track of the subgoals they are working on. When students make incorrect proof steps, EPIC explains their errors using the expert plans.

In addition to these ITSs, there are a number of other computer-based learning environments that are not thought of as ITSs, but that do provide explicit representations of experts' reasoning and representations. These include: the Writing Partner, which coaches students in the stages of writing and models the high-level, metacognitive questions that good writers ask themselves (Zellermayer, Salomon, Globerson & Givon, 1991); Dynagrams, which allows students to create and run simulations of problems in geometrical optics (Pea, 1992); and a simulation trainer for process control of a power plant described by Bennett (1992).

Coaching

Coaching consists of modeling, hints, feedback, reminders, and new tasks (Collins et al., 1989). The learning environments discussed in the previous subsection employ a range of different kinds of coaching. First, the modeling of expert's reasoning and representations is a form of instruction, even though the instruction may be implicit. For example, it may involve the use of an interface feature, such as an animated display of a word problem situation in Animate, or a template for an expert proof plan in EPIC. Second, most of these systems explicitly alert students whenever their performance differs from an expert. Third, some of the systems give additional feedback such as the criteria that were used to determine the student's error, and the correct action in that situation. For example, MACH III uses the information in the troubleshooting trees to critique students incorrect actions and explain better actions.

None of the learning environments discussed in the previous subsection do any student diagnosis beyond recording student actions. Thus, these systems cannot tailor their coaching to individual students' knowledge states. However, Newman (1989) suggests that human expert tutors in apprenticeship situations sometimes do not seem to make use of information about students' misconceptions. Rather, the experts simply point out when a student makes an error and, through hints or modeling, show the correct procedure. This is the approach to coaching followed by non-diagnostic ITSs.

Fading

To implement fading, a tutor gradually reduces the amount and explicitness of coaching as a student progresses, so that students are always able to solve instructional problems, and always contribute as much of their own thinking to a solution as is possible. Diagnostic ITSs probably do a better job than non-diagnostic tutors at fading, because they base decisions about when to fade out coaching on a student diagnosis that provides more specific, dynamic information about students' changing knowledge and misconceptions than non-diagnostic tutors.

The concept of fading is an answer to the general questions of when to provide instruction, and what kinds of instruction to provide. Shute (1992) has described the diagnostic approach to fading as *microadaptive instruction*. She has also suggested a non-diagnostic alternative to this approach, called *macroadaptive instruction*. In the microadaptive approach, one uses student diagnosis to make small-scale decisions about instructional interventions throughout the course of a tutoring session. In the macroadaptive approach, one uses assessments of student aptitudes prior to interaction with the ITS to make a one-time decision about what type of instructional approach the ITS will take with a student. The assessment of student aptitudes can be based on standardized tests or data from the first few minutes of a student's interaction with the ITS.

In one study, Shute (1992) tested students on a computerized battery of tests of associative learning ability. She then had them use an ITS that taught electricity laws. Each student used either a version of the ITS that took a direct-instruction approach, or one that took a guided-discovery approach. While there were no overall (main) effects of the direct-instruction or guided-discovery ITSs leading to better student performance, aptitude-treatment interactions were found, in which students with a particular aptitude were better suited to a particular instructional approach. For example, students with high associative-learning ability learned better in the guided-discovery ITS, while low associative-learning students fared better in the direct-instruction ITS. Shute has found similar aptitude-treatment interactions using other measures of student aptitude, such as early exploratory behavior by students using the electricity tutor (Shute, in press a), and other ITSs, such as an ITS for teaching flight engineering skills (Shute, in press b).

These results suggest that global measurements of student aptitudes can be used to choose the instructional approach an ITS should use with a student, and that choosing the right instructional approach for a student can increase learning. Since this macroadaptive approach to making instructional decisions does not involve student diagnosis, it is potentially less expensive than the microadaptive approach.

Articulation / Reflection

Articulation refers to the goal of teaching students to talk about the knowledge and skills they are learning. For example, Pea (1992) suggests that a crucial part of the task of learning science is learning to "talk science" (i.e., to participate in scientific discourse). Reflection involves getting students to compare their knowledge with that of an expert or another student.

The various means described above by which non-diagnostic tutors can explicitly model experts' representations and reasoning can help students reflect on their knowledge. Other ways that non-diagnostic tutors can use to encourage reflection include presenting replays and summaries of students' problem solving. Collins et al. (1992) advocate using "abstracted replay[s], in which the determining features of expert and student performance are highlighted." The replays and summaries of INCOFT, which were described above, provide just this kind of

information. MACH III and the Sherlock II (Katz & Lesgold, in press) also provide replays and summaries.

Both articulation and reflection are facilitated by a collaborative approach to learning. Katz and Lesgold describe a number of methods by which students can use ITSs collaboratively, including working together on a problem, posing problems to each other, and critiquing each others' solutions (using replays and summaries). Allowing students to work together on problems and critique each others' solutions gives them opportunities to explain their reasoning to each other. And, recent research suggests that students who explain their reasoning during learning learn more (Chi, Bassock, Lewis, Reimann & Glaser, 1989). In addition, allowing students to collaborate in a classroom setting, as was done with INCOFT, MACH III, and Dynagrams, gives the teacher the chance to guide students when their discussions get off track.

Another way to facilitate articulation and reflection using ITSs is to provide curriculum support materials to teachers. For example, the MACH III developers provided teachers with view graphs and posters that modeled expert knowledge (e.g., troubleshooting trees) and manuals of explanations (Kurland, et al., 1992). These were intended to facilitate classroom discussion of knowledge conveyed by the ITS.

To summarize, one of the most powerful ways in which non-diagnostic ITSs can improve learning is by offering devices, such as explicit displays of experts' representations and reasoning, and replays and summaries of students' performance, that can facilitate collaborative discussions among students and teachers. These discussions can help students to articulate and reflect on their knowledge.

Sequencing

Collins et al. suggest that learning tasks should be sequenced so that, over time, they increase in complexity, diversity, and specificity. The specificity criterion refers to the goal of teaching a whole task before focusing on specific subtasks. Diagnostic tutors provide more guidance on task sequencing, using the student model to determine how complex or specific a task to present next. The non-diagnostic approach is to leave sequencing decisions up to the teacher and the students.

Summary of Characteristics of Non-Diagnostic ITSs

This section has summarized characteristics of non-diagnostic ITSs such as explicit modeling of experts' knowledge, lack of student modeling, focus on student / teacher collaboration using replays and summaries, and using pre-assessments of student abilities to choose instructional environments. I have presented three examples of systems that have many of these characteristics and are characterized by their designers as ITSs -- INCOFT, MACH III, and EPIC. In addition, I have described four computer-based learning environments that also share the features of non-diagnostic ITSs but which are not described by their designers as ITSs -- Animate, Dynagrams, the Writing Partner, and a part-task trainer for process-control plants.

In Appendix A, each of these seven systems is described in terms of the low-tech and high-tech features used in Table 1. For comparison, the appendix also contains descriptions of the two diagnostic ITSs discussed above, the LISP Tutor and Sherlock I, and Sherlock II, a hybrid of diagnostic and non-diagnostic ITSs. The appendix shows that the category of non-diagnostic ITSs includes systems that are not commonly thought of as ITSs. Thus, considering the non-diagnostic approach expands our conception of how intelligence can be incorporated in computer-based learning environments.

Another benefit of the non-diagnostic approach mentioned earlier was that it can be used in converting black-box expert systems and simulations to instructional systems. In order to show an example of this, in the final section I will describe how the non-diagnostic approach is being used to develop an intelligent maintenance job aid into a training system.

IV. AN APPLICATION OF THE NON-DIAGNOSTIC APPROACH TO TRAINING SYSTEM DESIGN

The US Air Force's Armstrong Laboratory is developing the Integrated Maintenance Information System (IMIS), an intelligent job aid to help maintenance technicians perform aircraft maintenance on the flightline (Link et al., 1987). Technicians using IMIS will take a "portable maintenance aid" (a laptop-sized computer) to the aircraft that will provide the following kinds of information: suggestions of troubleshooting tests and replacements, displays of maintenance procedures and schematics, aircraft built-in-test results, aircraft history, and parts availability. Mei Technology Corporation is currently developing a prototype to show how IMIS can be used to train technicians in maintenance procedures and troubleshooting.

IMIS contains an expert system that provides troubleshooting suggestions (concerning tests and replacements). The troubleshooting algorithm used by the expert system is different from the reasoning of expert human troubleshooters in some important ways (Hicks, Gugerty, Young & Walsh, in press). For example, IMIS does not use a mental model of the malfunctioning system, and chooses tests and replacements by doing exhaustive calculations involving fault probabilities and test and replacement times for every component that could be causing a malfunction. Thus, IMIS is closer to a black-box than a glass-box expert system. This makes it difficult to use IMIS's expert system as the basis of a student diagnosis module.

Mei Technology is using non-diagnostic ITSs as models in converting IMIS to a training system, both because of IMIS's black-box expert module, and because of the educational potential of the non-diagnostic approach. The IMIS training system will give students an opportunity to practice maintenance procedures (including troubleshooting) in a realistic simulation environment. For example, in one version of the system, students will use the same portable maintenance aid that they would use on the aircraft. The training system will explicitly model expert troubleshooting reasoning by taking students through the steps of troubleshooting problems, as is done in the Writing Partner and EPIC. It will also explicitly model expert troubleshooting strategies such as elimination and half split, using displays of system block

diagrams. The training system will use the troubleshooting expert system to identify when a student suggests a bad test or replacement and provide coaching in the form of the information IMIS used to make its choice in that situation. Finally, the IMIS training system will facilitate articulation and reflection by using replays and summaries of students' troubleshooting sessions, in which students' actions are compared to IMIS's. Preliminary design work suggests that these non-diagnostic features can be added to IMIS to create a relatively low-cost training system, because extensive task analysis will not be necessary beyond that performed for developing the IMIS expert system.

V. CONCLUSION

Non-diagnostic ITSs offer a potentially fruitful approach to computer-based education and training that complements the approach taken by traditional diagnostic ITSs. The lack of student diagnosis in non-diagnostic tutors will likely result in lower tutor development costs. In addition, the non-diagnostic approach promises to have positive educational value. Non-diagnostic features such as modeling experts' representations, replaying and summarizing students' performance, and focusing on collaborative learning implement some of the key aspects of the successful cognitive-apprenticeship approach to training and education.

A number of computer-based training systems have been developed recently with non-diagnostic features, some that are explicitly identified as ITSs, and some that are not. The framework for describing ITSs presented in Table 1 provides a way of classifying both of these kinds of computer-based training systems, and comparing them to traditional, diagnostic ITSs. Thus this framework may help expand our conception of how to incorporate intelligence in computer-based training.

The non-diagnostic approach can be applied in the development of new ITSs and in converting existing systems (e.g., job aids and expert systems) to ITSs. I gave an example of how the IMIS maintenance job aid is being converted into a non-diagnostic ITS, and also mentioned that the Sherlock ITS, which was initially a traditional, stand-alone tutor, is incorporating non-diagnostic features such as performance replays so that it can be used to foster collaborative learning.

Before non-diagnostic ITSs can become widely used, however, a number of obstacles must be overcome. The first obstacle concerns how to integrate these ITSs into the classroom and train teachers to use them. Some of the problems encountered in the MACH III evaluation relate to these issues. For example, some instructors neglected to use the troubleshooting trees because they felt the trees gave too much help to the students, even though the developers and some students thought this feature of the ITS had significant instructional value. This example shows that when instructors see an ITS as not fitting their instructional methods and goals, they will tailor it, if possible, to fit. Although this tailoring could occur with any ITS, it is much more likely with non-diagnostic ITSs, because instructors are more closely involved in their use than with stand-alone, diagnostic tutors.

This phenomenon suggests the need for training instructors in how, and why, to use all the features of a non-diagnostic ITS. In fact, instructors in the MACH III felt they received insufficient training. In some cases, it may be necessary to assist instructors in moving from a lecturer role to a facilitator/coach role that fits better with using a non-diagnostic ITS.

The second obstacle to widespread use of non-diagnostic ITSs is the lack of empirical validation of their effectiveness. Conducting rigorous research that tests these systems in their intended educational settings (i.e., classrooms and other collaborative learning situations) should be a high priority.

Finally, the work conducted by Mei Technology has led to an approach using a personal computer (PC). The PC-based, non-diagnostic intelligent tutoring capability should link the diagnostic and training relevant information resident in the IMIS data bases to a simulated work environment. The situational analysis conducted by these reseachers revealed the potential for use of the maintenance ready room (where trainees review training documents before engaging in specific troubleshooting and training activities). By locating the multimedia capabilities of the PC in the ready room, it would be possible to use the Diagnostic Module and the Content Data Module to generate high fidelity training simulations. Another advantage to using the PC versus the PMA hardware platform would be the cost per unit; the PMA costs nearly 10 times more than the PC. In terms of training, the PC offers significant gains in terms of training fidelity with the use of video and audio which allow the trainee to literally watch the expert perform a maintenance task and hear the expert express his/her reasoning for this approach. The PC is a more versatile and proven platform for involving students in the learning process and the level of interactivity far exceeds what will be available on the PMA. The PC training capability should augment the PMA reserving work on the actual aircraft to required repairs and maintaining a high level of readiness to meet any mission requirement.

REFERENCES

Acchione-Noel, S., Saia, F.E., Williams, L., & Sarli, G. (1990). Maintenance aid computer hawk intelligent institutional instructor: Training developments study. (Report No. TRAC-WSMR-TEA-90-O51). White Sands Missile Range, NM: U.S. Army TRADOC Analysis Command.

Acker, L., Lester, J., Souther, A., & Porter, B. (1991). Generating coherent explanations to answer students' questions. In H. Burns, J. W. Parlett, & C. L. Redfield (Eds.), *Intelligent tutoring systems: Evolutions in design* (pp. 151-176). Hillsdale, NJ: Erlbaum.

Anderson, J. R. (1983). *The architecture of cognition.* Cambridge, MA: Harvard University Press.

Anderson, J. R. (1988). The expert module. In M. C. Polson & J. J. Richardson (Eds.), *Foundations of intelligent tutoring systems* (pp. 21-54). Hillsdale, NJ: Erlbaum.

Anderson, J. R., & Reiser, B. J. (April, 1985). The LISP tutor. *Byte*, pp. 159-175.

Anderson, J. R., Boyle, C. F., & Yost, G. (1985). The geometry tutor. In A. Joshi (Ed.), *Proceedings of the Ninth International Joint Conference on Artificial Intelligence.* Los Altos, CA: Morgan Kaufmann.

Anderson, J. R., Conrad, F. G., & Corbett, A. T. (1989). Skill acquisition and the LISP tutor. *Cognitive Science, 13*, 467-506.

Anderson, J. R., Corbett, A. T., Fincham, J. M., Hoffman, D., & Pelletier, R. (1992). General principles for an intelligent tutoring architecture. In J. W. Regian & V. J. Shute (Eds.), *Cognitive Approaches to automated instruction.* Hillsdale, NJ: Erlbaum.

Bennett, K. (1992). The use of on-line guidance, representation aiding, and discovery learning to improve the effectiveness of simulation training. In W. Regian & V. Shute (Eds.), *Cognitive approaches to automated instruction*. Hillsdale, NJ: Erlbaum.

Bonar, J. G. (1991). Interface architectures for intelligent tutoring systems. In H. Burns, J. W. Parlett, & C. L. Redfield (Eds.), *Intelligent tutoring systems: Evolutions in design*. Hillsdale, NJ: Erlbaum.

Burger, M. L., & DeSoi, J. F. (1992). The cognitive apprenticeship analogue: A strategy for using ITS technology for the delivery of instruction and as a research tool for the study of teaching and learning. *International Journal of Man-Machine Studies, 36*, 775-795.

Burns, H. L., & Capps, C. G. (1988). Foundations of intelligent tutoring systems: An introduction. In M. C. Polson & J. J. Richardson (Eds.), *Foundations of intelligent tutoring systems*. Hillsdale, NJ: Erlbaum.

Burton, R. R., & Brown, J. S. (1982). An investigation of computer coaching for informal learning activities. In D. Sleeman & J. S. Brown (Eds.), *Intelligent tutoring systems.* New York, NY: Academic Press.

Burton, R. R. (1988).The environment module of intelligent tutoring systems. In M. C. Polson & J. J. Richardson (Eds.), *Foundations of intelligent tutoring systems.* Hillsdale, NJ: Erlbaum.

Chi, M., Bassock, M., Lewis, M. Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, 13, 145-182.

Collins, A., Brown, J. S. & Newman, S., (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. Resnick (Ed.), *Knowing, learning and instruction.* Hillsdale, NJ: Erlbaum.

Corbett, A., & Anderson, J. (1991). Feedback control and learning to program with the CMU Lisp Tutor. Paper presented at the 72nd Annual Meeting of the American Educational Research Association, Chicago, IL.

Gery, G. (1991). *Electronic performance support systems: How and why to remake the workplace through the strategic application of technology.* Boston, MA: Weingarten Publications.

Halff, H. M. (1988). Curriculum and instruction in automated tutors. In M. C. Polson & J. J. Richardson (Eds.), *Foundations of intelligent tutoring systems.* Hillsdale, NJ: Erlbaum.

Hicks, K., Gugerty, L., Young, W. & Walsh, W. (in press). Training situation analysis for flightline maintenance training: Implications for IMIS. Armstrong Laboratory Technical Paper. San Antonio, TX: Brooks Air Force Base.

Katz, S., & Lesgold, A. (1992). Modeling the student in Sherlock II. Paper presented at the International Joint Conference on Artificial Intelligence Workshop on Agent Modeling for Intelligent Interaction.

Katz, S., & Lesgold, A. (in press). The role of the tutor in computer-based collaborative learning situations. In S. Lajoie & S. Derry (Eds.), *Computers as cognitive tools.* Hillsdale, NJ: Erlbaum.

Kurland, L., Granville, R., & MacLaughlin, D. (1992). Design, development, and implementation of an intelligent tutoring system for training radar mechanics to troubleshoot. In M. Farr & J. Psotka (Eds.), *Intelligent instruction by computer.* Philadelphia, PA: Taylor & Francis.

Lajoie, S. P., & Lesgold, A. (1989). Apprenticeship training in the workplace: Computer-coached practice environment as a new form of apprenticeship. *Machine-Mediated Learning, 3*, 7-28.

Lesgold, A., Eggan, G., Katz, S., & Rao, G. (1992). Possibilities for assessment using computer-based apprenticeship environments. In W. Regian & V. Shute (Eds.), *Cognitive approaches to automated instruction*. Hillsdale, NJ: Erlbaum.

Link, W. R., Von Holle, J. C., & Mason, D. (1987). Integrated Maintenance Information System (IMIS): A maintenance information delivery concept. (Report No. AFHRL-TP-87-27). Dayton, Ohio: Wright-Patterson AFB.

Means, B., & Gott, S. P. (1988). Cognitive task analysis as a basis for tutor development: Articulating abstract knowledge representations. In J. Psotka, L. D. Massey, & S. A. Mutter (Eds.), *Intelligent tutoring systems: Lessons learned*. Hillsdale, NJ: Erlbaum.

Merrill, D., Reiser, B., Ranney, M., & Trafton, J. G. (1992). Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. *The Journal of the Learning Sciences*.

Nathan, M., Johl, P., Kintsch, W., & Lewis, C. (1989). An unintelligent tutoring system for solving word algebra problems. In D Bierman, J. Breuker & J. Sandberg (Eds.), *Proceedings of the fourth international conference on artificial intelligence and education*. Springfield, VA: IOS.

Nathan, M., Kintsch, W., & Young, E. (1992). A theory of algebra word problem comprehension and its implications for the design of learning environments. *Cognition and Instruction, 9*(4), 329-390.

Newman, D. (1989). Apprenticeship or tutorial: Models for interaction with an intelligent instructional system In *Proceedings of the Eleventh Annual Meeting of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum.

Newman, D., Grignetti, M., Gross, M., & Massey, L. D. (1989). Intelligent Conduct of Fire Trainer: Intelligent technology applied to simulator-based training. *Machine-Mediated Learning, 3*, 29-39.

Pea, R. (1992). Augmenting the discourse of learning with computer-based learning environments. In E. de Corte, M. Linn, H. Mandl & L. Verschaffel (Eds.), *Computer-based learning environments and problem solving* (NATO Series ASI Series F). New York, NY: Springer Verlag.

Pirolli, P. (1991). Computer-aided instructional design systems. In H. Burns, J. W. Parlett, & C. L. Redfield (Eds.), *Intelligent tutoring systems: Evolutions in design*. Hillsdale, NJ: Erlbaum.

Riel, M., Levin, J., & Miller-Souviney, B. (1987). Learning with interactive media: Dynamic support for students and teachers. In R. Lawler & M. Yazdani (Eds.), *Artificial intelligence and education (Volume One): Learning environments and tutoring systems*. Norwood, NJ: ABLEX.

Schumacher, E. (1973). *Small is beautiful*. New York, NY: Harper & Row.

Shute, V. (1991). How to build more intelligent tutoring systems. *Proceedings of the 33rd Annual Conference of the Military Testing Association.* San Antonio, TX.

Shute, V. (1992). Aptitude-treatment interactions and cognitive skill diagnosis. In W. Regian & V. Shute (Eds.), *Cognitive approaches to automated instruction.* Hillsdale, NJ: Erlbaum.

Shute, V. (in press a). A comparison of learning environments: All that glitters... In S. Lajoie & S. Derry (Eds.), *Computers as cognitive tools.* Hillsdale, NJ: Erlbaum.

Shute, V. (in press b). A macroadaptive approach to tutoring. *Journal of Artificial Intelligence and Education.*

Towne, D. M., & Munro, A. (1988). The Intelligent Maintenance Training System. In J. Psotka, L. D. Massey, & S. A. Mutter (Eds.), *Intelligent tutoring systems: Lessons learned.* Hillsdale, NJ: Erlbaum.

Towne, D. M., & Munro, A. (1991). RAPIDS: A simulation-based instructional authoring system for technical training. (Report No. AFHRL-TR-90-49). San Antonio, TX: Brooks Air Force Base.

Towne, D. M., & Munro, A. (1992). Supporting diverse instructional strategies in a simulation-oriented training environment. In J. W. Regian & V. J. Shute (Eds.), *Cognitive Approaches to automated instruction.* Hillsdale, NJ: Lawrence Erlbaum Ass.

Twidale, M. (1989). Intermediate representations for student error diagnosis and support. In D Bierman, J. Breuker & J. Sandberg (Eds.), *Proceedings of the fourth international conference on artificial intelligence and education.* Springfield, VA: IOS.

VanLehn, K. (1988). Student modeling. In M. C. Polson & J. J. Richardson (Eds.), *Foundations of intelligent tutoring systems.* Hillsdale, NJ: Erlbaum.

Wiederholt, B. J., Browning, E. J., Norton, J. E., & Johnson, W. B. (1991). *MITT Writer and MITT Writer advanced development: Developing authoring and training systems for complex technical domains.* Atlanta, GA: Galaxy Corp. Prepared for Armstrong Laboratory Human Resources Directorate (HRTI), Brooks AFB, TX.

Woods, D., Roth, E., & Bennett, K. (1991). Explorations in joint human-machine cognitive systems. In S. Robertson, W. Zachary & J. Black (Eds.), *Cognition, computing, and cooperation.* Norwood, NJ: ABLEX.

Woolf, B. (1991). Representing, acquiring, and reasoning about tutoring knowledge. In H. Burns, J. W. Parlett, & C. L. Redfield (Eds.), *Intelligent tutoring systems: Evolutions in design.* Hillsdale, NJ: Erlbaum.

Zellermayer, M., Salomon, G., Globerson, T., & Givon, H. (1991). Enhancing writing-related metacognitions through a computerized writing partner. *American Educational Research Journal, 28* (2), 373-391.

APPENDIX A: FEATURES OF NON-DIAGNOSTIC AND DIAGNOSTIC ITSs

System: **INCOFT**
ITS Type: Non-diagnostic
Reference: Newman, D., Grignetti, M., Gross, M., & Massey, L. D. (1989). Intelligent Conduct of Fire Trainer: Intelligent technology applied to simulator-based training. *Machine-Mediated Learning, 3*, 29-39.

	Low-Tech			High-Tech
Expert Module				
Human like reasoning ?	no expert module		black-box expert	glass-box expert
Type of solutions	none		canned	generated on-line
Diagnostic Module				
Type of student model	none	performance data	issued-based diagnosis	model tracing
Use of student model	none	replays & summaries		to choose instruction
Instructional Module				
Curricular decisions	teacher		preset by developer	generated on-line
Instructional method decisions	teacher		preset by developer	generated on-line
Instructional intervention decisions	teacher		preset by developer	generated on-line
Group or individual use?	collaborative		stand-alone	both
Interface				
Realistic ?	not at all		to some extent	yes
Models expert ?	very little		somewhat	extensively

System: **MACH III**
ITS Type: Non-diagnostic
Reference: Kurland, L., Granville, R. & MacLaughlin, D. (1992). Design, development, and implementation of an intelligent tutoring system for training radar mechanics to troubleshoot. In M. Farr & J. Psotka (Eds.), *Intelligent instruction by computer.* Philadelphia, PA: Taylor & Francis.

	Low-Tech			High-Tech
Expert Module				
Human like reasoning ?	no expert module		$\boxed{\text{black-box expert}}$	glass-box expert
Type of solutions	none		$\boxed{\text{canned}}$	generated on-line
Diagnostic Module				
Type of student model	none	$\boxed{\text{performance data}}$	issued-based diagnosis	model tracing
Use of student model	none	$\boxed{\text{replays \& summaries}}$		to choose instruction
Instructional Module				
Curricular decisions	$\boxed{\text{teacher}}$		preset by developer	generated on-line
Instructional method decisions	$\boxed{\text{teacher}}$		preset by developer	generated on-line
Instructional intervention decisions	$\boxed{\text{teacher}}$		preset by developer	generated on-line
Group or individual use?	$\boxed{\text{collaborative}}$		stand-alone	both
Interface				
Realistic ?	not at all		$\boxed{\text{to some extent}}$	yes
Models expert ?	very little		somewhat	$\boxed{\text{extensively}}$

System: **Animate**
ITS Type: Non-diagnostic
Reference: Nathan, M., Johl, P., Kintsch, W. & Lewis, C. (1989). An unintelligent tutoring system for solving word algebra problems. In D Bierman, J. Breuker & J. Sandberg (Eds.), *Proceedings of the fourth international conference on artificial intelligence and education.* Springfield, VA: IOS.

	Low-Tech			High-Tech
Expert Module				
Human like reasoning ?	$\boxed{\text{no expert module}}$		black-box expert	glass-box expert
Type of solutions	$\boxed{\text{none}}$		canned	generated on-line
Diagnostic Module				
Type of student model	$\boxed{\text{none}}$	performance data	issued-based diagnosis	model tracing
Use of student model	$\boxed{\text{none}}$	replays & summaries		to choose instruction
Instructional Module				
Curricular decisions	$\boxed{\text{teacher}}$		preset by developer	generated on-line
Instructional method decisions	$\boxed{\text{teacher}}$		preset by developer	generated on-line
Instructional intervention decisions	$\boxed{\text{teacher}}$		preset by developer	generated on-line
Group or individual use?	collaborative		$\boxed{\text{stand-alone}}$	both
Interface				
Realistic ?	not at all		$\boxed{\text{to some extent}}$	yes
Models expert ?	very little		somewhat	$\boxed{\text{extensively}}$

System: **Dynagrams**
ITS Type: Non-diagnostic
Reference: Pea, R. (1992). Augmenting the discourse of learning with computer-based learning environments. In E. de Corte, M. Linnn, H. Mandl & L. Verschaffel (Eds.), *Computer-based learning environments and problem solving* (NATO Series ASI Series F). New York, NY: Springer Verlag.

	Low-Tech			High-Tech
Expert Module				
Human like reasoning ?	no expert module		black-box expert	glass-box expert
Type of solutions	none		canned	generated on-line
Diagnostic Module				
Type of student model	none	performance data	issued-based diagnosis	model tracing
Use of student model	none	replays & summaries		to choose instruction
Instructional Module				
Curricular decisions	teacher		preset by developer	generated on-line
Instructional method decisions	teacher		preset by developer	generated on-line
Instructional intervention decisions	teacher		preset by developer	generated on-line
Group or individual use?	collaborative		stand-alone	both
Interface				
Realistic ?	not at all		to some extent	yes
Models expert ?	very little		somewhat	extensively

System: **EPIC**
ITS Type: Non-diagnostic
Reference: Twidale, M. (1989). Intermediate representations for student error diagnosis and support. In D Bierman, J. Breuker & J. Sandberg (Eds.), *Proceedings of the fourth international conference on artificial intelligence and education.* Springfield, VA: IOS.

	Low-Tech			High-Tech
Expert Module				
Human like reasoning ?	no expert module		$\boxed{\text{black-box expert}}$	glass-box expert
Type of solutions	none		$\boxed{\text{canned}}$	generated on-line
Diagnostic Module				
Type of student model	$\boxed{\text{none}}$	performance data	issued-based diagnosis	model tracing
Use of student model	$\boxed{\text{none}}$	replays & summaries		to choose instruction
Instructional Module				
Curricular decisions	$\boxed{\text{teacher}}$		preset by developer	generated on-line
Instructional method decisions	teacher		$\boxed{\text{preset by developer}}$	generated on-line
Instructional intervention decisions	teacher		$\boxed{\text{preset by developer}}$	generated on-line
Group or individual use?	$\boxed{\text{collaborative}}$		stand-alone	both
Interface				
Realistic ?	not at all		$\boxed{\text{to some extent}}$	yes
Models expert ?	very little		somewhat	$\boxed{\text{extensively}}$

System: **Writing Partner**
ITS Type: Non-diagnostic
Reference: Zellermayer, M., Salomon, G., Globerson, T. & Givon, H. (1991). Enhancing writing-related metacognitions through a computerized writing partner. *American Educational Research Journal, 28* (2), 373-391.

	Low-Tech			High-Tech
Expert Module				
Human like reasoning ?	no expert module		black-box expert	glass-box expert
Type of solutions	none		canned	generated on-line
Diagnostic Module				
Type of student model	none	performance data	issued-based diagnosis	model tracing
Use of student model	none	replays & summaries		to choose instruction
Instructional Module				
Curricular decisions	teacher		preset by developer	generated on-line
Instructional method decisions	teacher		preset by developer	generated on-line
Instructional intervention decisions	teacher		preset by developer	generated on-line
Group or individual use?	collaborative		stand-alone	both
Interface				
Realistic ?	not at all		to some extent	yes
Models expert ?	very little		somewhat	extensively

System: **Process-plant control simulation with on-line advisor**
ITS Type: Non-diagnostic
Reference: Bennett, K. (1992). The use of on-line guidance, representation aiding, and discovery learning to improve the effectiveness of simulation training. In W. Regian & V. Shute (Eds.), *Cognitive approaches to automated instruction*. Hillsdale, NJ: Erlbaum.

	Low-Tech			High-Tech
Expert Module				
Human like reasoning ?	no expert module		black-box expert	glass-box expert
Type of solutions	none		canned	generated on-line
Diagnostic Module				
Type of student model	none	performance data	issued-based diagnosis	model tracing
Use of student model	none	replays & summaries		to choose instruction
Instructional Module				
Curricular decisions	teacher		preset by developer	generated on-line
Instructional method decisions	teacher		preset by developer	generated on-line
Instructional intervention decisions	teacher		preset by developer	generated on-line
Group or individual use?	collaborative		stand-alone	both
Interface				
Realistic ?	not at all		to some extent	yes
Models expert ?	very little		somewhat	extensively

System: **LISP Tutor**
ITS Type: Diagnostic
Reference: Anderson, J. R., & Reiser, B. J. (April, 1985). The LISP tutor. *Byte*, pp. 159-175.

	Low-Tech			High-Tech
Expert Module				
Human like reasoning ?	no expert module		black-box expert	glass-box expert
Type of solutions	none		canned	generated on-line
Diagnostic Module				
Type of student model	none	performance data	issued-based diagnosis	model tracing
Use of student model	none	replays & summaries		to choose instruction
Instructional Module				
Curricular decisions	teacher		preset by developer	generated on-line
Instructional method decisions	teacher		preset by developer	generated on-line
Instructional intervention decisions	teacher		preset by developer	generated on-line
Group or individual use?	collaborative		stand-alone	both
Interface				
Realistic ?	not at all		to some extent	yes
Models expert ?	very little		somewhat	extensively

System: **Sherlock I**
ITS Type: Diagnostic
Reference: Lajoie, S. P & Lesgold, A. (1989). Apprenticeship training in the workplace: Computer-coached practice environment as a new form of apprenticeship. *Machine-Mediated Learning, 3*, 7-28.

	Low-Tech			High-Tech
Expert Module				
Human like reasoning ?	no expert module		$\boxed{\text{black-box expert}}$	glass-box expert
Type of solutions	none		$\boxed{\text{canned}}$	generated on-line
Diagnostic Module				
Type of student model	none	performance data	$\boxed{\text{issued-based diagnosis}}$	model tracing
Use of student model	none	replays & summaries		$\boxed{\text{to choose instruction}}$
Instructional Module				
Curricular decisions	teacher		preset by developer	$\boxed{\text{generated on-line}}$
Instructional method decisions	teacher		$\boxed{\text{preset by developer}}$	generated on-line
Instructional intervention decisions	teacher		$\boxed{\text{preset by developer}}$	$\boxed{\text{generated on-line}}$
Group or individual use?	collaborative		$\boxed{\text{stand-alone}}$	both
Interface				
Realistic ?	not at all		$\boxed{\text{to some extent}}$	yes
Models expert ?	very little		$\boxed{\text{somewhat}}$	extensively

System: **Sherlock II**

ITS Type: Hybrid of diagnostic and non-diagnostic

Reference: Katz, S., & Lesgold, A. (1992). Modeling the student in Sherlock II. Paper presented at the International Joint Conference on Artificial Intelligence Workshop on Agent Modeling for Intelligent Interaction.

	Low-Tech			High-Tech
Expert Module				
Human like reasoning ?	no expert module		black-box expert	glass-box expert
Type of solutions	none		canned	generated on-line
Diagnostic Module				
Type of student model	none	performance data	issued-based diagnosis	model tracing
Use of student model	none	replays & summaries		to choose instruction
Instructional Module				
Curricular decisions	teacher		preset by developer	generated on-line
Instructional method decisions	teacher		preset by developer	generated on-line
Instructional intervention decisions	teacher		preset by developer	generated on-line
Group or individual use?	collaborative		stand-alone	both
Interface				
Realistic ?	not at all		to some extent	yes
Models expert ?	very little		somewhat	extensively